PARTITION TREE FOR A SEGMENTATION-BASED VIDEO CODING SYSTEM

M. Pardàs, P. Salembier, F. Marqués and R. Morros

Dept. of Signal Theory and Communications
ETSETB - Universitat Politècnica de Catalunya
C/ Gran Capità, 08034 Barcelona, Spain
E-mail: montse@gps.tsc.upc.es

ABSTRACT

In this paper we describe a very low bit rate video coding system that belongs to the class of object-based coding systems. The system takes advantage of homogeneity in both gray level and motion, trying to use the most adequate criterion for defining a partition in order to obtain the best compromise between cost and quality. In order to do so, multiple partitions are generated for every image in a hierarchical way, from the coarsest one to the finest one. Then, a decision step will chose which parts of the image need to be coded with regions from one partition or from another. The aim of the paper is to study the features that this multiple partition must have, and propose a way to create it.

1. INTRODUCTION

Object-based video coding systems try to take advantage of the redundancy in the images due to the fact that every object will have, in general, an homogeneity either in gray level or in motion. In this way, higher compression ratios can be obtained if the images are processed using these homogeneous entities, than working on a pixel basis or with square blocks, as it is usual in classical coding systems. These entities will then be processed in terms of contours, textures and motion. So, the first aim is to obtain a partition of the image in objects which can be easily described with these features.

Currently, a large number of works are focused on the definition of the best partition in order to take advantage of these redundancies [3]. In [12], only the gray level homogeneity is taken as object feature and used to define the partition. It is expected to find a partition which corresponds to the real objects of the scene and so, hopefully, they will have coherent motion. Moreover, there is a tracking of regions which defines the time evolution of the regions. This allows content based functionalities [1] as well as a region based motion estimation for the coding [1], [6]. In [5], motion redundancy is used to create the partition but gray level homogeneity is not taken into account.

In [9] a new scheme for object based coding systems was introduced. In this system, a joint optimization is performed for every frame, in order to find its optimum partition, together with a representation model associated to each resulting image segment. This optimization is based on the theory of optimal resources allocation [2]. The optimization is performed on an scenario which comprises several possible partitions provided, together with a given set of available models. In [9], the partitions for every frame were generated with a purely spatial method, recursively partitioning the frames into variable size cells along a quadtree structure. In general, a number of partitions has to be generated, instead of a unique partition as in [12] or [5].

In this paper, we introduce the concept of *Partition Tree*, a hierarchical partition structure which can be used in systems which perform an optimization over partitions and coding techniques. We also propose, for the generation of such a *Partition Tree*, a method based in [8], which allows region tracking of regions along the sequence. Moreover, the *Partition Tree* allows to introduce, in the generation of the final partition, gray-level as well as motion criteria.

2. GENERAL DESCRIPTION OF THE SYSTEM

The whole process can be described in three steps, as shown in Figure 1: Partition proposal, Decision and Coding [1].

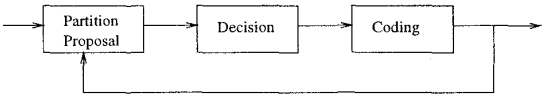

Figure 1. General scheme of the coding system

The aim of the first step is to generate a number of partition from which the partition to be coded must be selected. We will analyze in the next section the features of this block.

The decision step has to select the partition to be coded, together with the coding tools most appropriated for every region in the partition. For this optimization, an strategy based on the rate-distorsion theory is used.

Finally, the coding step has to code all the necessary information to synthesize the images at the receiver end. This implies, basically, contour, texture and motion information.

3. PARTITION PROPOSAL

The partition proposal has to generate a number of partitions among which the decision will select the regions to code. Let us see the main features that such partitions should have:

- Hierarchical structure. In order to have an efficient generation of partitions, the decision should be able to choose regions belonging to different partitions for building the partition to code. The best solution is to create different partitions with different resolution in such a way that every level is a refinement of the previous level. That is, every partition must contain the same regions than the one in the previous level (with the same contours), and it has to add some new regions in order to refine it. This will allow to choose different resolution levels for the different objects of the image. Furthermore, it allows an efficient optimization in the decision part. We call this hierarchical structure a *Partition Tree*.

- Object based. In order to get a good homogeneity in texture and in motion inside the regions produced, the partitions should be based, as much as possible, on the objects shape.

- Region tracking along the sequence. In order to take advantage of the time redundancy in the coding part, a motion

0-7803-3192-3/96 $5.00©1996 IEEE

compensation based on regions should be used. Also, content based functionalities demanded in new video services, need to define the time evolution of the regions. For these reasons, purely spatial segmentation techniques should be discarded.

In next section we describe the system which we propose in order to generate a *Partition Tree* with the features mentioned above.

4. PARTITION TREE GENERATION

It is composed of three main steps: Projection, Merging and Segmentation. The objective of the first block is to ensure the time evolution of the regions of the partition. The time evolution of the partition at time T is defined using the partition at time T-1 and the original frames T-1 and T. Although this is a purely region tracking step, the projected partition is, in general, a rather good approximation of the optimal partition of the current frame. However, some new objects may have appeared in the scene and new regions may need to be introduced. On the other hand, several regions belonging to the same object can be processed globally because, for example, they have the same motion. In this case, these regions can be merged to create one single object and to decrease the coding cost devoted to the partition. The Segmentation and Merging blocks deal with these possible modifications of the partition topology: the introduction of new regions (Segmentation) and the elimination of useless regions (Merging). Thus, the *Partition Tree* is formed by the projected partition plus a set of hierarchical partitions that are "above" and "below" the projected partition, as illustrated in Figure 2.

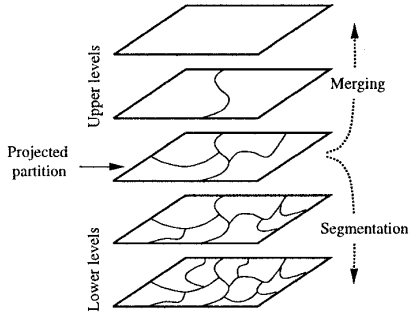

Figure 2. Partition Tree representation

4.1. Projection

The partition projection adapts the partition P_{T-1} of frame T-1 to the current frame T without introducing new regions. Note that partition P_{T-1} is the partition chosen by the decision for representing the previous image. In the projection algorithm, the region correspondence problem is solved using a connectivity criterion [8]. This restriction works when dealing with regions undergoing simple deformations or small motion. If this is not the case, the positions of an object in two successive frames may become disconnected. In order to solve this problem, and correctly project moving regions, the motion is estimated and used in the projection [7]. Therefore the partition projection is performed in two steps: motion estimation and compensation and extension of the previous partition into the current frame.

Motion is estimated between the frame which has just been segmented F_{T-1} and the current frame F_T. A classical block matching algorithm is used. The motion vectors obtained are applied to the original image as well as to the previous partition, thus obtaining a compensated partition Pc_{T-1} where, hopefully, corresponding regions are time-connected. Therefore, the extension of the partition into the current frame can be performed with a connectivity-based algorithm.

The algorithm used to perform this extension is the classical morphological segmentation tool called Watershed [4]. Given a set of markers defining the regions of interest, this algorithm finds the precise contours of these regions. Markers are connected components with a specific label identifying the presence of homogeneous regions. These markers serve as seeds for a region growing process: to each marker will correspond one region, and all regions together will tessellate the space. In order to use this algorithm for the extension of the partition of the previous image into the current one, two 3D signals are constructed. Frames Fc_{T-1} and F_T are grouped together to form a temporal block $\mathcal{F}$ of size 2 in the time dimension. Similarly, the frame Pc_{T-1} is grouped with an empty frame P_o representing an entire frame of uncertainty. The resulting 3D signal ($\mathcal{P}$) is considered as the set of markers that should be used to segment the signal $\mathcal{F}$. The marker extension is achieved with the watershed algorithm applied on these 3D signals. The watershed extends the markers defined by Pc_{T-1} into the empty frame P_o. Each pixel of the uncertainty area (that is, of image P_o) is assigned to a region of image Pc_{T-1} based on a similarity criterion.

Finally, two remarks must be taken into account. First, the watershed algorithm has to be implemented without using the gradient images, as it has to work with 3D signals [11]. Second, the similarity criterion used considers the gray level difference between pixels but also the contour complexity (spatial as well as temporal contours) in order to achieve contours which are easy to code and time stability [8].

4.2. Segmentation

From the projected partition the other levels of the *Partition Tree* are built. As mentioned, lower levels are created by re-segmenting the regions of the projected partition. This provides the possibility of obtaining new regions which were not present in the previous one. The reasons for obtaining these new regions are the twofold. Firstly, if new objects in the scene are introduced, new regions must be created. Note that new objects appearing in the scene which should be coded as independent regions will be characterized by their texture, which is different from that of their neighboring objects. Secondly, two regions with a very different texture but with the similar motion may have been merged in the past; if suddenly they adopt divergent motions, a large compensation error will be created if they continue to be coded with the same motion parameters. In this case too, it might be worth separating the two regions again and code two different sets of motion parameters.

In both cases, we need to separate regions with inhomogeneous texture. Therefore the re-segmentation procedure which will produce the lower levels of the *Partition Tree* is based on a texture criterion.

This segmentation is implemented with a hierarchical structure which progressively introduces new regions in the partition at every level without modifying the contours of the previous levels. It starts from the projected partition and it can be described in four different steps:

-Image modeling and extraction of the residue. Each level has to improve the segmentation of the previous level by introducing new significant regions. These improvements should apply to regions which cannot be easily coded due to large changes of texture in their inside. To get information about these regions, each projected region is coded with a simple model. The difference between the coded and the original images, called the Modeling Residue, is then computed. The Modeling Residue concentrates all the information about the regions of the previous partition which contain inhomogeneous texture. Therefore, this signal is

used to drive the segmentation in the following steps.

- Image simplification. The Modeling Residue is simplified in order to eliminate the information which is not relevant for the extraction of new regions in the current hierarchical step. Different simplification tools are used depending on the simplification criterion. Morphological operators are very interesting in this step, because they allow us to simplify images with visually important criteria. For instance if the simplification aims at removing regions smaller than a given limit, area filters can be used [13]. If the goal is to keep regions with a contrast higher than a given value, then h-maxima and h-minima operators can be used [10] . In any case, the simplification should yield areas of almost constant gray level values (flat zones), in order to make the following steps easier.

- Marker extraction. The goal of this step is to detect, in the simplified Modeling Residue, the presence of relevant regions. For each relevant region, a marker is set, identifying some pixels belonging to this region. The implementation of this step depends on the simplification used previously. In the case of the morphological operators mentioned before, it consists on a simple flat zones labeling procedure [10].

- Region boundary decision. After marker extraction, the number and the core of the regions to be segmented are known. Then, the watershed algorithm is used to assign every pixel of the uncertainty areas between markers to one of these markers, based on a similarity criterion. As for the projection, the similarity criterion used for the watershed takes into account the gray level difference as well as the contours complexity.

In the *Partition Tree*, typically, size criterion is used for the first re-segmentation levels and contrast criterion is used for the last one. At each level, the same procedure is repeated, computing the residue with the partition of the previous level, and the only difference is the simplification degree which is decreased to progressively introduce small or low-contrasted regions. This segmentation procedure is also used to produce the first partition in the intra frame mode. That is, starting from a segmentation with one region (the whole image), this hierarchical procedure obtains a segmentation with a predefined number of regions. From this partition, the *Partition Tree* is built, merging regions in the upper levels and segmenting them in the lower ones.

4.3. Merging

On the upper levels of the *Partition Tree*, coarser partitions are created. As the contours in the higher levels must also be present in the lower ones, the partitions above the projected one are created by merging regions. This merging is based on a motion criterion. The aim is to merge neighboring regions which have a similar motion. When these regions can be compensated using the same motion parameters, the contour between them does not need to be coded and only one set of parameters is needed.

To perform the merging for a given level, the motion parameters of its regions must have already been computed. That is, we need to know the motion parameters which relate every region of the partition in the previous image with the partition in the current one. For the projected level there is an extensive motion estimation, where the affine motion parameters for every region are found [1]. With these motion parameters the merging of the projected level is performed. Then, there is a motion refinement to find the motion parameters for the regions of the first merged level (above the projected level), which allows to compute the next merged level.

Once the motion parameters of a given level are known, the estimation of the cost of merging neighboring regions can begin. For every couple of neighboring regions, a merging cost is computed. Then, the required number of merged regions will be obtained by selecting the couples of neighboring regions with the smaller merging costs.

The merging of regions is performed taking into account that the regions of the partition obtained can be coded by means of motion compensation and coding of the prediction error, if chosen in the Decision block. So, the merging criterion must be related with the cost of coding the prediction error. However, the actual decision of merging two regions in the precise terms of coding cost will be taken in the decision process. In this merging procedure several merging proposals must be produced at every level. For these proposals, only the compensation error produced by the merged regions, which is closely related to the coding cost, is taken into account.

Thus, the cost which has been considered is the increase of the mean square compensation error in the merged region with respect to the mean square compensation error when the two regions are compensated separately. Once this cost has been computed for every couple of neighboring regions, the merging proposals will be made selecting the smaller merging costs.

In the case of intra frames, the motion is not taken into account for producing the merging, as the coding does not use any motion information. The merging is made on a texture basis. The procedure is the same, but the costs are computed taking into account the difference of the mean gray level value of the neighboring regions.

5. DECISION AND CODING

Once the *Partition Tree* has been created, the Decision block has to select the best coding strategy in terms of regions and coding techniques among a set of possibilites. The *Partition Tree* represents the whole set of possibilities concerning the regions and a set of texture coding techniques are proposed to the decision. This last set involves several region-based coding techniques with various levels of quality. Moreover, the techniques can be proposed in intra mode and in inter mode.

Figure 3 summarizes the overwhole decision process: from the em Partition Tree, all regions are extracted to form the *Decision Tree*. Several texture coding techniques are considered for each region. Then, taking regions from various levels of the partition tree, the Decision block defines the final partition and assigns the best coding technique to each region.

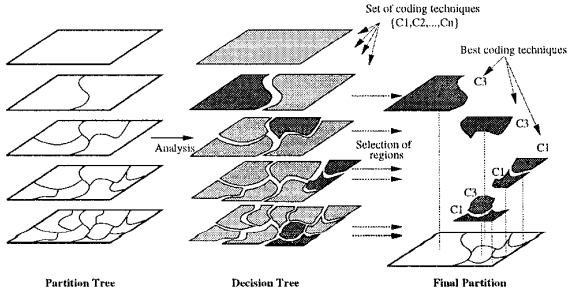

Figure 3. Decision process

Once the optimum partition and the coding strategy have been chosen, the information necessary to decode the sequence should be sent to the receiver. This information is composed of the coding strategy itself, the motion parameters for the regions that should be compensated, the partition and, finally, the texture parameters of each region.

Figure 4. Example of inter-frame segmentation

6. RESULTS

This coding system has been used to code several video sequences, leading to a good quality for very low bit rates. As an example we show frame 150 of the sequence Foreman coded at 33 Kbit/s (in QCIF format, using a frame rate of 5 frames/s and coding the three color components). For this image we show, in the first row, the partition corresponding to the previous image. In second row we can see the partition proposals for the current frame. The central picture in the *Partition Tree* corresponds to the projection of the partition of the previous image. We can observe, in the *Partition Tree* how regions with similar motion are merged, and the progressive increase in the resolution. In the third row we can see the original image, the selected partition and the coded image. As it can be observed, the selected partition contains regions belonging to the different levels of the *Partition Tree*.

7. CONCLUSIONS

In this paper we have introduced the concept of *Partition Tree*, a segmentation structure which can be used in a new type of object based coding systems.

The main features of this structure are its hierarchical character and its region tracking capability. An implementation to construct a *Partition Tree* has also been proposed. In this implementation the time continuity is achieved with a projection step based on the temporal connectivity of the regions. The other levels of the *Partition Tree* are constructed by merging regions with a similar motion or segmenting the regions which have inhomogeneous texture.

REFERENCES

[1] I. Corset et al. Segmentation-based coding system allowing manipulation of objects. Technical Report 408, ISO/IEC JTC1/SC29/WG11, MPEG 95, 1995.

[2] H. Everett. Generalized lagrange multiplier method for solving problems of optimum allocation of resources. *Operation Res.*, 11:399–417, 1963.

[3] F. Marqués, M. Pardàs, and P. Salembier. Coding-oriented segmentation of video sequences. In L. Torres and M. Kunt, editors, *Video Coding: The Second Generation Approach*. Kluwer, 1996.

[4] F. Meyer and S. Beucher. Morphological segmentation. *Journal of Visual Communication and Image Representation*, 1(1):21–46, 1990.

[5] H.G. Musmann, Hőtter, and J. M. Ostermann. Object-oriented analysis-synthesis coding of moving images. *Signal Processing, Image Comm.*, 1(2):117–138, 1989.

[6] M. Pardàs and P. Salembier. 3D morphological segmentation and motion estimation for image sequences. *EURASIP Signal Processing*, 38(2):31–43, 1994.

[7] M. Pardàs and P. Salembier. Joint region and motion estimation with morphological tools. In *Second Workshop on Mathematical Morphology and its Applications to Signal Processing*, pages 93–100, France, 1994.

[8] M. Pardàs and P. Salembier. Time-recursive segmentation of image sequences. In *EUSIPCO 94*, pages 18–21, Edinburgh, U.K., 1994.

[9] E. Reusens. Joint optimization of representation model and frame segmentation for generic video compression. *EURASIP Signal Processing*, 46(11):105–117, 1995.

[10] P. Salembier. Morphological multiscale segmentation for image coding. *EURASIP Signal Processing*, 38(3):359–386, 1994.

[11] P. Salembier and M. Pardàs. Hierarchical morphological segmentation for image sequence coding. *IEEE Trans. on Image Processing*, 3(5):639–651, 1994.

[12] P. Salembier, L. Torres, F. Meyer, and C. Gu. Region-based video coding using mathematical morphology. *Proceedings of IEEE*, 83(6):843–857, 1995.

[13] L. Vincent. Grayscale area openings and closings, their efficient implementation and applications. In *First Workshop on Mathematical Morphology and its Applications to Signal Processing*, pages 22–27, Spain, 1993.